A New Method to Stop Spam Emails in Sender Side

R. Kholghi[1], E. Eidkhani[1], S.A. Mortazavi[1], M. Hajyvahabzadeh[1], A. Nemaney Pour[2]
[1]Sharif University of Technology, International Campus/Dept. of IT Engineering, Kish Island, IRAN
Email: {r_kholghi, elina, anahita, melisa}@kish.sharif.edu
[2]J-TECH Corporation/ R&D Section, Yokohama, Japan
Email: a.nemaney@jtechno.jp

***Abstract*— Nowadays one of the major problems by Internet users, who they have email addresses, are undesirable emails (also known as *spam*). Spam emails generally with profitable reasons are sent to the large number of email addresses. A spammer, who sends spam, tries to run an advertisement for companies or products. The problem with these spams is that they waste the network resources. In this paper a method is presented to stop spam emails in the sender side. In this method, the sender mail server checks any email based on some pre-defined criteria. If the sender mail server determines that the email is not spam, it will deliver that mail to associated mail server. Otherwise, the email is blocked in the sender mail server. In this method the waste of network resources such as time, allocated memory, and bandwidth are preserved.**

Index Terms— filtering, sender mail server, receiver mail server, spam email, user license.

I. Introduction

Today, one of the most popular and efficient way for communication in the world is electronic mail (email). By increasing the use of this communication media, some problems have appeared. Receiving the unwanted emails is the major problem that email owners do not have any interest in the content of such emails. These unwanted emails are known as spam emails. Spam emails are generated in a large scale and sent to the large number of email addresses by a spammer. Spam emails not only waste the network resources, they cause security damages to the system as well. According to the recent reports, 50% of the email traffic through the Internet includes spam emails [1]. Many solutions have been proposed to detect and stop spam emails. The common issue with all of them is that they have tried to stop the spam emails at the receiver side [2-6]. It means that the spam detection is performed after the mail is delivered to the receiver mail server. This approach by itself wastes the network resources when the received email is a spam one. In this paper we propose a mechanism to detect and stop the spam emails before transferred to the receiver side. This paper is prepared as follows: In section II some related works are discussed. Section III shows our proposed method. In Section IV the structure of our filtering method is shown. Section V contains the comparisons of our method with other methods. Finally, the conclusion is given in section VI.

II. Related Work

To prevent spam emails, some techniques such as reverse-lookup and filtering methods [2-6] have been proposed. In reverse lookup [2], before any transactions, a session should establish between sender and the relevant receiver. The sender introduces him/herself to the receiver by sending "hello" packet. Then, the receiver performs the reverse lookup operation through a reliable DNS server. If the returned value from the DNS server is equal to the sender's FQDN (Fully Qualified Domain Name), the receiver gives the permission for sending an email to the sender. Otherwise, the sender is determined as a spammer and he/she is not permitted to send his/her email. The problem of this method is about the mobile users and also the users without valid IP address. Therefore, the users with correct emails but different IP addresses cannot send their emails. The other suggested techniques are types of filtering method [3-5]. There are many types of filtering such as blacklist [3], whitelist [4], content based filtering such as Bayesian filter, and rule based filtering such as Spamassassin [5]. In blacklist filtering, the emails with the IP addresses in the blacklist are blocked. On the other hand, the emails with the IP addresses in the whitelist are trustable. These methods help the users to classify their emails. The key difficulty with these lists is updating the related lists. Bayesian filtering searches all the words in a message to find out some illegitimate words. At the beginning, the user should have a database to add the words in two parts, legitimate and illegitimate words. In the next steps, the Bayesian filter based on its database can recognize whether the email is spam or not. One of the difficulties with content based filtering is that, the user needs to train the related filters. In rule based filtering some patterns such as exclamation point, unambiguous words and etc. are used. The advantage of this filtering is that they can be easily installed in compare with Bayesian filtering. However, the key problem is that, it is probable that valid emails are recognized as spam ones.

III. Proposed Method

In this section, we present the principles of our proposed method for preventing the spam emails. As stated before, the purpose of this proposal is to find a way to preserve network resources. For this purpose, instead of concentrating and performing the filtering techniques on the receiver mail server, we do the necessary actions on the sender mail server. The main goal is to detect the spam emails at the source (sender side), and to avoid sending them through the network. The following steps are clarifying our proposed method as shown in Fig. 1:

(1) A session needs to be set up. When a sender wants to send an email, in the first step he/she should connect to his/her mail server. This step is called the *connection phase*.

(2) The authentication phase should be done by sender mail

DOI: 01.IJNS.02.04.43

server. This step is for detecting the valid users. In fact, the user should have an email address supported by the mail server. We call this phase as *authentication phase*.
(3) The sender should upload the email to his/her mail server. This step is done if in the authentication phase has recognized the user as a valid one. Later we will refer to this step as *uploading phase*.
(4) The mail server based on its criteria decides to send the email or not. If the email is determined as a spam one, it will be blocked; if not, it will be sent to the relevant mail server. This step is called *decision phase*.
(5) Finally, if the email is distinct as a correct one (ham), it will be sent to the receiver mail server. We titled this step as *distribution phase*.

IV. Detial Structure of Proposed Method

This section presents the details of our proposal. As stated before, for delivering an email to the specific destination some steps are necessary. Here, these steps are explained in details.

A. Connection Phase

As Fig. 2 shows, at the first step, when the sender wants to connect to his/her mail server, The IP address of the relevant mail server needs to be lookup. This is done by a query to DNS (Domain Name System). After that, the DNS server responses back the requested IP address to the sender. After that, the sender sends a "*Hello*" packet to the mail server to establish a connection (here we call it *session*) through three-way-handshake. To accomplish the three-way-handshake process, the sender sends a synchronisation message to his/her mail server. Then, his/her mail server sends the synchronisation acknowledge and also the session agreement. At this time the sender and the mail server are connected and the server starts the authentication phase.

B. Authentication Phase

In this phase, the sender mail server sends "OK" message as a confirmation. This means that the server accepts the session. After that, the sender sends his/her email address via the "USER" command to the mail server. The mail server checks the email address with its database. If the email address is not valid, the mail server sends a notification indicating that the email address does not exist. Otherwise, the mail server sends the "ACK" message.

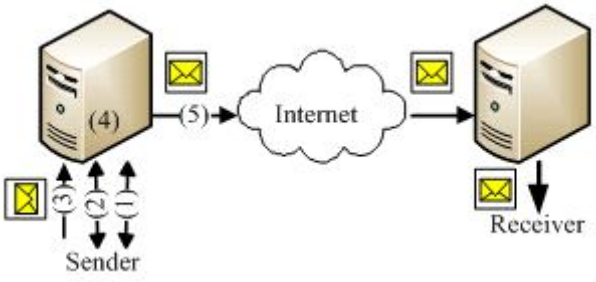

Figure1. Overall view of our proposed method

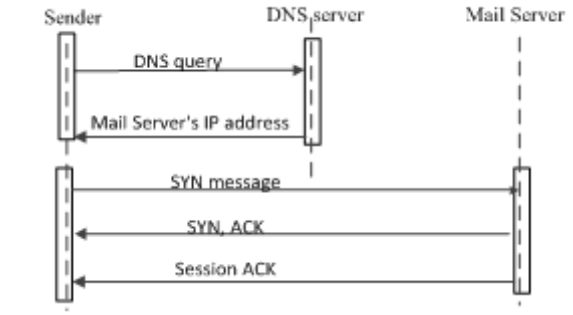

Figure 2. Connection phase

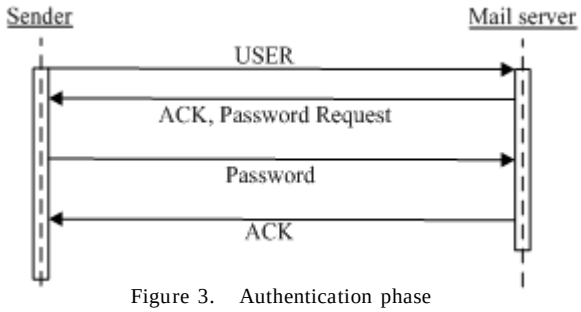

Figure 3. Authentication phase

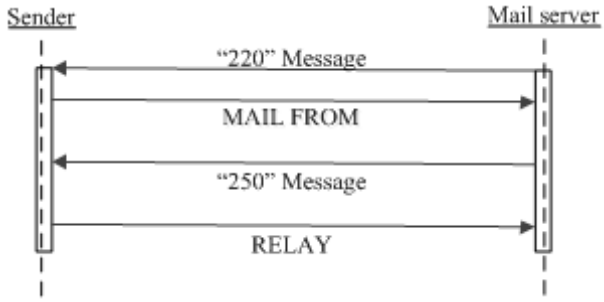

Figure 4. E-mail Uploading phase

This shows the confirmation for the client email address. Afterwards, the mails server also requests for the client password. In the next transmission, the client transfers his/her password to the mail server. The mail server checks the client password. In this step also if the password is correct, the mail server sends an "ACK" message to the sender. Otherwise, the mail server sends the failure message to the sender. Normally, we have this message from the server," the password is wrong. Please enter a right password". At this time, the authentication phase is done successfully. Fig. 3 illustrates the orders of these commands. Now the client is allowed to upload his/her email to the mail server.

C. Uploading Phase

In uploading phase, the sender uploads his/her email. The mail server sends "220" message (220 is a code) which means that the mail server gets ready for starting a communication with the sender. The sender sends his/her email by "*MAIL FROM*" command. This command is a confirmation for the "220" message. At this time, the sender's email is uploaded to his/her allocated space in the mail server. After finishing uploading process, the mail server sends "250" message to the sender. Then, the sender gives the email address of the receiver through a "*RELAY*" command to the mail server. Notice that, the email can be sent to one or more

DOI: 01.IJNS.02.04.43

receivers. Fig. 4 illustrates the sequence of uploading phase.

D. Decision Phase

In this phase, the mail server should decide on sending or blocking the email based on some predefined criteria such as previously proposed filtering methods. Obviously, the mail server has the main role for this routine. In our proposed method, the suggestion is to use anti-spam methods in sender mail server. With this idea we can save the network and receiver resources. Fig. 5 shows the mail server decision policies as follows. These policies are suggested in a routine to make challenges in sender side for sending bulk emails. So, if a sender is a spammer, it is hard for him/her to send spam emails through the network.

a) Checking the Sender User-licence

The first parameter which has to be checked is the sender's user-licence as a threshold for each user [6]. User-licence is the pre-defined number of outgoing emails for each user in an interval time. In other words, the user-licence is the number of email addresses sent by a sender in an interval. If the amount of receivers is more than his/her pre-defined threshold, the mail server makes an interrupt in the process of sending those emails.

b) Checking the Sender Whitelist

The second parameter has to be checked is the whitelist. For this purpose, the mail server compares the receivers' email addresses with the existing ones in the whitelist. At the first step, if the receiver's email address does not exist in the sender's whitelist, the mail server announces the sender to add that address to his/her whitelist. Formerly, the mail server checks the existence of that new email address in the sender whitelist. This process makes some challenges for the spammer to distribute spam emails because if the spammer cannot add such addresses to the whitelist, so they are prevented from sending spam emails.

c) Applying the Filtering Methods

When the mail server assures that all of the receiver email addresses are added to the sender's whitelist, then the filtering techniques are applied. The mail server uses different kinds of filtering methods to filter the spam emails. One of the major filtering methods is checking the subject of the email. In this method, the mail server has a word list which contains the common words that spammers use them usually in their emails headers. If any such words are detected, the E-mail will be blocked. In addition, the commercial spam emails include some links and/or specific words in the body. Commonly, the mail server by saving the list of these words and links can determine these kinds of spam emails too. This procedure is known as another filtering technique. Finally, content based and rule based filtering may be applied to detect spam emails.

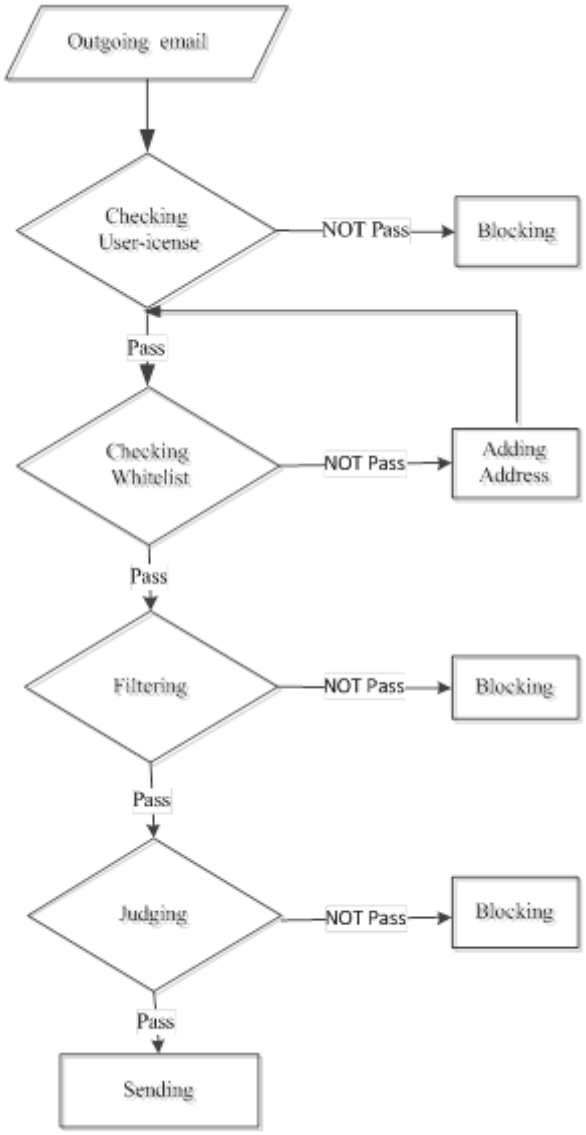

Figure 5. Decision phase flowchart

d) The Judgement on the Other Mail Servers opinions

Sometimes, a spam email may pass all of the previous limitations. In this case, the mail server should have a judgement based on the report from the other mail servers. For each user, the mail server maintains a report about the users' behavior received from the other mail servers. If the email sender was known as a spammer, then the mail server blocks his/her emails.

e) Distribution Phase

When the sender mail server determines that the email is not a spam, the email is prepared to be distributed to the destinations in the network.

V. Comparision

In this section, we compare our proposed method with the current spam mail detection method. Fig. 6 shows the result of spam mail detection in the sender side compared with the one in the receiver side. For this experiment, we used

DOI: 01.IJNS.02.04.43

TREC[1] as open source anti-spam software. The experiment was performed on *1000* emails, to compute the processing time for spam email detection. The results show that the time required to filter *1000* emails in the receiver side is 120 seconds, whereas in the similar situation the time required to filter the same number of emails takes just 0.6 second. Consequently, filtering in the sender side takes decreases the processing time significantly. The major difference in our proposed method is that spam filtering process is occurred in the source. By filtering in the sender mail server, if any sender tries to send spam emails, the mail server blocks it at each step before proceeding to the next ones. Consequently, the spam sender is the main victim of his/her crime. According to our method, every user needs to add all the email addresses to his/her whitelist before sending the emails to the network. Because of this property, when a spammer abuses the account of a valid user, the mail is blocked and the spammer is requested to add that address to the whitelist. If a spammer cannot access the whitelist, he cannot send a spam email. As a result, it is not desirable for the spammers to add all the receiver addresses to the whitelist. Therefore, the spammers have serious challenges for sending spam emails. Since our anti-spam method, especially filtering, operates on the sender side, the expected number of filtering operations will be decreased. For example, if a user sends a spam email to one-hundred users in the network, by using the other methods the filtering operation should be done one-hundred times in each receiver system separately. But by using our method, this amount is reduced to one. In the proposed method, if a sender decides to distribute lots of spam emails in the network, whenever one of the other mail servers recognizes that email as a spam, it informs the sender mail server as soon as possible. Now, the relevant mail server knows that sender as a spammer and blocks the user email account immediately. So, the spammer will be unable to send his/her remaining spam emails to the receivers.

VI .Conclusion

In this paper, we have proposed a new method for stopping spam emails. Our proposed method has tried to solve the spam emails distribution problem in the sender side whereas the previous proposed methods have presented solutions in the receiver side. In this method, whenever a spam email is detected, the mail server will block it. The key idea of this method is blocking spam emails in the source. As a result, the processing cost of numerous servers, network resources and time will be saved more. Furthermore, this method tries to prevent to send spam email through the network by making challenges for malicious users. The major result of this method is that the transmission cost of spam emails is diverted to the sender system.

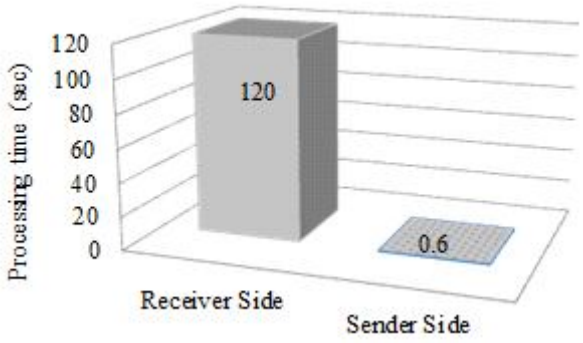

Figure6. Evaluation of spam mail detection in sender and receiver side

References

[1] D. Nagamalai, B.C. Dhinakaran, and J.K. Lee, "An In-depth Analysis of Spam and Spammers," *International Journal of Security and its Applications*, vol. 2, No.2, pp. 9-22, April 2008.

[2] B. Agrawal, N. Kumar, and M. Molle, "Controlling spam emails at the routers," *Proceeding of IEEE International Conf. on Communications (ICC 2005)*, Seoul, Korea, pp. 1588-1592, May 2005.

[3] A. Ramachandran, D. Dagon, and N. Feamster, "Can dns-based blacklists keep up with bots?," *The Third Conference on Email and Anti-Spam (CEAS 2006)*, California, USA, pp. 1-2, Jul. 2006.

[4] A.Ramachandran, and N. Feamster, "Understanding the network-level behavior of spammers," *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2006)*, Pisa, Italy, pp. 291-302, Sep. 2006.

[5] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C.D. Spyropoulos, and P. Stamatopoulos, "Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach," *4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000)*, Lyon, France, pp.1-13, Sep. 2000.

[6] R. Kholghi, S. Behnam Roudsari, and A. Nemaney Pour, "An Efficient Spam Mail Detection by Counter Technique," *International Conference on Computer Science and Information Technology (ICCSIT 2011)*, Penang, Malaysia, pp. 497-500, Feb. 2011.

[1] TREC: Text Retrieval Conference. http://plg.uwaterloo.ca/

DOI: 01.IJNS.02.04. 43